

WEST Search History

DATE: Tuesday, February 13, 2007

Hide?	<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>
	<i>DB=PGPB,USPT,USOC; PLUR=NO; OP=OR</i>		
<input type="checkbox"/>	L27	l26 and ((display\$ or interfac\$ or view\$) with (second adj1 (processor or cpu or computer or terminal or device)))	4
<input type="checkbox"/>	L26	l25 and ((display\$ or interfac\$ or view\$) with (first adj1 (processor or cpu or computer or terminal or device)))	11
<input type="checkbox"/>	L25	(l23 or l24) and ((file or files or folder or folders) near (browser\$ or code or program\$ or software))	786
<input type="checkbox"/>	L24	707/104.1.ccls.	5900
	<i>DB=PGPB; PLUR=NO; OP=OR</i>		
<input type="checkbox"/>	L23	707/1.ccls.	3193
	<i>DB=PGPB,USPT,USOC; PLUR=NO; OP=OR</i>		
<input type="checkbox"/>	L22	L11 and L21	64
<input type="checkbox"/>	L21	(display\$ with computers with (file or files or folder or folders))	609
<input type="checkbox"/>	L20	(help adj1 desk).ab.	36
<input type="checkbox"/>	L19	((single adj1 display) with (first adj1 computer))	1
<input type="checkbox"/>	L18	L12 and (display\$ with ((file or files or folder or folders) adj1 systems))	39
<input type="checkbox"/>	L17	L11 and (display\$ with (computer adj1 systems))	64
<input type="checkbox"/>	L16	L11 and (display\$ with ((file or files or folder or folders) adj1 systems))	14
<input type="checkbox"/>	L15	L11 and (display\$ with (first near second near computer\$))	11
<input type="checkbox"/>	L14	L11 and "computer-1"	3
<input type="checkbox"/>	L13	L11 and (computer near 1)	462
<input type="checkbox"/>	L12	(file or files or folder or folders).ab.	35869
<input type="checkbox"/>	L11	(file or files or folder or folders).ti.	12745
<input type="checkbox"/>	L10	(multidimensional near brows\$)	7
<input type="checkbox"/>	L9	L2 and (file adj1 systems)	14
<input type="checkbox"/>	L8	(6757739 6560608 20040177058).pn.	3
<input type="checkbox"/>	L7	(6757739 6560608).pn.	2
<input type="checkbox"/>	L6	(5835089 20050120084).pn.	2
<input type="checkbox"/>	L5	L3 and ((display\$ or view\$) with (computer adj1 systems))	28
<input type="checkbox"/>	L4	browser.ab.	5463
<input type="checkbox"/>	L3	browser.ti.	1017

DB=PGPB; PLUR=NO; OP=OR

10/737,055

<input type="checkbox"/>	L2	browser.ti.	597
<input type="checkbox"/>	L1	20040078358.pn.	1

END OF SEARCH HISTORY

Dial g DataStar

options

logout

feedback

help

databases

easy
search

Advanced Search:


Inspec - 1898 to date (INZZ)

limit

Search history:

No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	single NEAR (display OR view OR interfac\$)	unrestricted	5784	show titles
2	INZZ	1 AND (display\$ OR view\$ OR interfac\$) NEAR (folder\$ OR file\$)	unrestricted	22	show titles



[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)
Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping
 whole document 

 Information added since: or: none 
 (YYYYMMDD)

search

☐ Documents with images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings 
- ☒ Inspec thesaurus - enter a term 
- ☒ Classification codes A: Physics, 0-1
- ☒ Classification codes A: Physics, 2-3
- ☒ Classification codes A: Physics, 4-5
- ☒ Classification codes A: Physics, 6
- ☒ Classification codes A: Physics, 7
- ☒ Classification codes A: Physics, 8

10/1137, 055

Dial g DataStar

options

logout

feedback

help

databases

search
page

titles

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

next
documents

save

locally as: PDF document



search strategy: do not include the search strategy



order

copy to
Clipboard☒ Select All

- 1 RRS: replica registration service for data grids.
- 2 Liquid 2D scatter space for file system browsing.
- 3 Concept and evaluation of X-NAS: a highly scalable NAS system.
- 4 A LabVIEW/sup R/ library for reading HP Chemstation files: pedagogical tools fo
- 5 The universal inbox.
- 6 Building user-interface for heterogeneous network device simulation: experien
- 7 Flexibility and performance of parallel file systems.
- 8 Flexibility and performance of parallel file systems.
- 9 Adding a data visualization tool to DEC FUSE.
- 10 libscheme: Scheme as a C library.
- 11 Database management in C++.
- 12 Multiprocessor file system interfaces.
- 13 A display processor.
- 14 A look at remote procedure calls.
- 15 A concurrent file system for a highly parallel mass storage subsystem.
- 16 Tools for remote computing in accelerator control.
- 17 Electronic charts-at the heart of the 21st century navigation.
- 18 An empirical study of software interface faults-an update.
- 19 A distributed UNIX system based on a virtual circuit switch.
- 20 The effect of airborne contamination and environmental conditions on the head/

☒ document 1 of 22 [Order Document](#)**Inspec - 1898 to date (INZZ)****Accession number & update**

0008865940 20070101.

Title

RRS: replica registration service for data grids.

Conference information

Data Management in Grids. First VLDB Workshop, DMG 2005. Revised Selected Papers, Trondheim, Norway, 2-3 Sept. 2005.

Source

Data Management in Grids. First VLDB Workshop, DMG 2005. Revised Selected Papers (Lecture Notes in Computer Science Vol. 3836), 2005, p. 100-12, 13 refs, pp. x+142, ISBN: 3-540-31212-9.
 Publisher: Springer-Verlag, Berlin, Germany.

Author(s)

Shoshani-A, Sim-A, Stockinger-K.

Editor(s): Pierson-J-M.



Author affiliation

Shoshani, A., Sim, A., Stockinger, K., Comput. Res. Div., California Univ., Berkeley, CA, USA.

Abstract

Over the last few years various scientific experiments and grid projects have developed different catalogs for keeping track of their data **files**. Some projects use specialized **file** catalogs, others use distributed replica catalogs to reference **files** at different locations. Due to this diversity of catalogs, it is very hard to manage **files** across grid projects, or to replace one catalog with another. In this paper we introduce a new grid service called the replica registration service (RRS). It can be thought of as an abstraction of the concepts for registering **files** and their replicas. In addition to traditional **single file** registration operations, the RRS supports collective **file** registration requests and keeps persistent registration queues. This approach is of particular importance for large-scale usage where thousands of **files** are copied and registered. Moreover, the RRS supports a set of error directives that are triggered in case of registration failures. Our goal is to provide a **single uniform interface** for various **file** catalogs to support the registration of **files** across multiple grid projects, and to make grid clients oblivious to the specific catalog used.

Descriptors

 FILE-ORGANISATION;  GRID-COMPUTING.

Classification codes

C6150N Distributed-systems-software*;

C6120 File-organisation.

Keywords

replica-registration-service; data-grids; **file-catalogs**; grid-service; **file-registration**.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper.

Publication year

2005.

Publication date

20050000.

Edition

2006015.

Copyright statement

Copyright 2006 The Institution of Engineering and Technology.

(c) 2007 The Institution of Engineering and Technology

☒ **document 2 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0008624279 20070101.

Title

Liquid 2D scatter space for **file** system browsing.

Conference information

Proceedings. Ninth International Conference on Information Visualisation, London, UK, 6-8 July 2005.

Source

Proceedings. Ninth International Conference on Information Visualisation, 2005, p. 451-6, 16 refs, pp. xxii+1011, ISBN: 0-7695-2397-8.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

Author(s)

Waldeck-C.

Editor(s): Banissi-E, Sarfraz-M, Roberts-J-C, Loften-B, Ursyn-A, Burkhard-R-A, Lee-A, Andrienko-G.

Author affiliation

Waldeck, C., Dept. Z3 (MIV: Mobile Inf. Visualization), ZGDV - Comput. Graphics Center, Darmstadt, Germany.

Abstract

This (video-) paper describes a user **interface** concept, which facilitates multidimensional visual queries, filtering and browsing of the **file** system at the same time by means of the "liquid scatter space" (LSS) concept (Waldeck, 2004). The UI concept is based on an advanced star field **display** (Ahlberg & Shneiderman, 1994) using liquid browsing (Waldeck, 2004) and selection based (sketch-) queries. It allows realtime search and browsing at the same time (integrated into one **single user interface**) and can visualize the most important **file** system meta data dimensions simultaneously: 1. **filename**; 2. creation /modification date; 3. **filesize**; 4. **filetype**; and 5. label. LSS provides very fast and easy visual data mining possibilities for the desktop and makes it possible to perceive complex dependencies between the most important **file** system meta data properties at a glance. It also overcomes the "hidden-deep-down-in-hierarchy-structures"-problem by making it possible to use **folders** for structuring but not being bound to them. This paper focuses on visual design and interaction aspects and emphasizes the importance of paying regard to visual interactive details for information visualization and interaction **interfaces** and the need of making it easy to use.

Descriptors

 DATA-MINING;  DATA-VISUALISATION;  FILE-ORGANISATION;  QUERY-PROCESSING;
 USER-INTERFACES.

Classification codes

C6120 File-organisation*;
C6130B Graphics-techniques;
C6180 User-interfaces.

Keywords

liquid-2D-scatter-space; **file-system-browsing**; **user-interface**; multidimensional-visual-queries; **file-system-filtering**; liquid-scatter-space-concept; realtime-search; realtime-browsing; visual-data-mining; information-visualization; **interaction-interfaces**.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper.

Availability

CCCC: 1550-6037/2005/\$20.00.

Publication year

2005.

Publication date

20050000.

Edition

2005043.

Copyright statement

Copyright 2005 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 3 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0007653004 20070101.

Title

Concept and evaluation of X-NAS: a highly scalable NAS system.

Conference information

Proceedings 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies,
San Diego, CA, USA, 7-10 April 2003.

Sponsor(s): IEEE Mass Storage Syst. Tech. Committee; NASA Goddard Space Flight Center.

Source

Proceedings 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies,
2003, p. 219-27, 10 refs, pp. xi+300, ISBN: 0-7695-1914-8.

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

Author(s)

Yasuda-Y, Kawarnoto-S, Ebata-A, Okitsu-J, Higuchi-T.

Abstract

X-NAS (expandable network attached storage), a highly scalable, distributed **file** system designed for entry-level NAS, has been developed. It virtualizes multiple NAS systems into a **single-file-** system **view** for different kinds of clients. The core of X-NAS is a multi-protocol virtualized **file** system (MVFS), and its key features - a smart-code wrapper daemon, **file-group** mapping, and a **file-handle** cache - improve X-NAS scalability. X-NAS has other key features for improving the manageability on many NAS systems; namely, on-line reconfiguration, autonomous rebalancing, and automatic migration, in which **files** are migrated automatically and dynamically independently of **file-sharing** services for clients. To validate the X-NAS concept, an X-NAS prototype was designed and tested according to the NFSv2 implementation. These tests indicate that X-NAS attains a quicker response time and higher throughput than a conventional **single** NAS, so its cost-performance scalability is also higher.

Descriptors

 CLIENT-SERVER-SYSTEMS;  NETWORK-OPERATING-SYSTEMS;  STORAGE-MANAGEMENT.

Classification codes

C6120 File-organisation*;

C6150N Distributed-systems-software.

Keywords

expandable-network-attached-storage; **highly-scalable-distributed-file-** system; X-NAS; **single-file-system-view**; multiple-systems; multi- **protocol-virtualized-file-system**; smart-code-wrapper-daemon; **file-** group-mapping; **file-handle-cache**; on-line-reconfiguration; autonomous-rebalancing; automatic-migration; clients; response-time; throughput; cost-performance-scalability.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper.

Availability

CCCC: 0-7695-1914-8/03/\$17.00.

Other format availability: Also available on CD-ROM in PDF format.

Digital object identifier

10.1109/MASS.2003.1194859.

Publication year

2003.

Publication date

20030000.

Edition

2003022.

Copyright statement

Copyright 2003 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at [USPTO Full Text Retrieval Options](#) [open url](#)

☒ **document 4 of 22** [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0005970564 20070101.

TitleA LabVIEW/sup R/ library for reading HP Chemstation **files**; pedagogical tools for GC-MS.**Source**

Laboratory Robotics and Automation, {Lab-Robot-Autom-USA}, 1998, vol. 10, no. 2, p. 119-21, 2 refs,
CODEN: LRAUEY, ISSN: 0895-7533.
Publisher: Wiley, USA.

Author(s)Starn-T.**Author affiliation**

Starn, T., Dept. of Chem., West Chester Univ., PA, USA.

Abstract

A LabVIEW/sup R/ program library that can access the chromatographic and mass spectrometric data stored in Hewlett-Packard (HP) Chemstation data **files** is described. The programs in the library can be used to extract total ion chromatograms or selected ion chromatograms. More importantly, a program is provided that will **display** an intensify graph of all mass values for every time in the chromatogram. This **display** of the data from a **single file** is used to illustrate the quantitative and qualitative power of gas chromatography-mass spectrometry (GC-MS), a so-called hyphenated an analytical chemistry technique.

Descriptors

CHROMATOGRAPHY; MASS-SPECTROSCOPY; SOFTWARE-LIBRARIES;
SPECTROCHEMICAL-ANALYSIS; SPECTROSCOPY-COMPUTING.

Classification codes

C7320 Physics-and-chemistry-computing*;
C6110J Object-oriented-programming.

Keywords

LabVIEW-library; **HP-Chemstation-files**; Hewlett-Packard; mass-spectrometry; ion-chromatograms;
gas-chromatography; analytical-chemistry.

Treatment codesP Practical.**Language**

English.

Publication typeJournal-paper.**Availability**

SICI: 0895-7533(1998)10:2L:119:LLRC; 1-#.
CCCC: 0895-7533/98/020119-03.

Publication year

1998.

Publication date

19980000.

Edition

1998028.

Copyright statement

Copyright 1998 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at

[custom url](#)[USPTO Full Text Retrieval Options](#)[open url](#)☒ **document 5 of 22** [Order Document](#)**Inspec - 1898 to date (INZZ)****Accession number & update**

0005700395 20070101.

Title

The universal inbox.

Source

BYTE (International Edition), {BYTE-Int-Ed-USA}, Sept. 1997, vol. 22, no. 9, p. 75-8, 80, 0 refs,
CODEN: BYTEDJ, ISSN: 0360-5280.
Publisher: McGraw-Hill, USA.





Author(s)

Hurwicz-M.

Abstract

To users assaulted from all sides by messages, a "universal inbox" sounds like a life saver. A universal inbox is a **single** user **interface**, usually based on e-mail client software, that lists all incoming messages. These messages can include e-mail, voice mail, fax mail (faxes that come through a fax server and arrive as e-mail), and more. From the universal inbox **interface**, the user can read, delete, or **file** all types of messages. The same **interface** inevitably contains functions for responding to messages too, although outbound messaging is not an inbox function. The universal inbox simplifies life by defragmenting the messaging environment. It's faster to check one inbox than multiple e-mail accounts, fax machines, and voice mail systems. Training requirements also diminish since you learn only one **interface**. The universal inbox may also eliminate delays in message reception that are due to the user's not checking a particular source (such as an e-mail account where you seldom receive any mail) frequently enough. The universal inbox can certainly simplify life-but not for all users. Although vendors have been working on universal inboxes for years, the technology is still maturing, so products might lack features you want. Moreover, while the universal inbox provides a **single** front end for disparate message types, it may do little to integrate administration, directories, or data stores on the back end. In addition, you must consider what you need and anticipate needing in the future.

Descriptors

 ELECTRONIC-MAIL;  FACSIMILE;  USER-INTERFACES;  VOICE-MAIL.

Classification codes

B6210G Electronic-mail*;
B6210D Telephony;
B6210H Facsimile-transmission;
C7104 Office-automation*;
C6180 User-interfaces.

Keywords

universal-inbox; **single-user-interface**; e-mail-client-software; incoming-messages; voice-mail; fax-mail; fax-server; inbox-function; messaging-environment; e-mail-accounts; training-requirements; message-reception; **single-front-end**; disparate-message-types.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 0360-5280(199709)22:9L:75:UI; 1-W.
CCCC: 0360-5280/97/\$1.50.

Publication year

1997.

Publication date

19970900.

Edition

1997038.

Copyright statement

Copyright 1997 IEE.

(c) 2007 The Institution of Engineering and Technology

Inspec - 1898 to date (INZZ)**Accession number & update**

0005527169 20070101.

TitleBuilding **user-interface** for heterogeneous network device simulation: experiences with MINIMOS.**Conference information**

Proceedings 1996 IEEE Hong Kong Electron Devices Meeting, Hong Kong, 29 June 1996.

Sponsor(s): IEEE Electron Devices Soc; Dept. Electron. Eng., Hong Kong Polytech. Univ.

Source

Proceedings. 1996 IEEE Hong Kong Electron Devices Meeting (Cat. No.96TH8146), 1996, p. 68-71, 5 refs, pp. ii+105, ISBN: 0-7803-3091-9. Publisher: IEEE, New York, NY, USA.

Author(s)Niu-G-F, Chen-R-M-M, Ruan-G.**Author affiliation**

Niu, G.F., Chen, R.M.M., Dept. of Electron. Eng., City Univ. of Hong Kong, Kowloon, Hong Kong.

Abstract

This paper describes **user-interface** building for device simulation using a heterogeneous computer network. After profiling MINIMOS and analyzing the user demands, we decide to paralyze the input parameter sets and then build the Multiple **Input-files** Multiple Parameter-sets (MIMP) **user-interface** and **Single Input-file** Multiple Parameter-sets (SIMP) **user-interface**. Both **interfaces** allow device engineers to exploit existing network resource to simulate multiple device parameter sets concurrently with minimal change to the validated standard codes, thus achieving low cost and reliable speedup.

Descriptors

DIGITAL-SIMULATION; ELECTRONIC-ENGINEERING-COMPUTING; MOSFET;
PARALLEL-PROGRAMMING; SEMICONDUCTOR-DEVICE-MODELS; USER-INTERFACES.

Classification codes

B2560R Insulated-gate-field-effect-transistors*;
B2560B Semiconductor-device-modelling-and-equivalent-circuits;
C7410D Electronic-engineering-computing*;
C6110P Parallel-programming;
C6180 User-interfaces;
C6185 Simulation-techniques.

Keywords

user-interface; heterogeneous-network-device-simulation; MINIMOS- MOSFET-simulator; input-parameter-sets; **multiple-input-files-multiple-** parameter-sets; **MIMP-interface**; **single-input-file-multiple-parameter-** sets; **SIMP-interface**.

Treatment codesP Practical.**Language**

English.

Publication type

Conference-paper.

Digital object identifier

10.1109/HKEDM.1996.566312.

Publication year

1996.

Publication date

19960000.

Edition

1997011.

Copyright statement

Copyright 1997 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 7 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0005464787 20070101.

Title

Flexibility and performance of parallel **file** systems.

Conference information

Proceedings of Third International Conference of the ACPC Parallel Databases and Parallel I/O, Klagenfurt, Austria, 23-25 Sept. 1996.

Source

Parallel Computation. Third International ACPC Conference with Special Emphasis on Parallel Databases and Parallel I/O Proceedings, 1996, p. 1-11, 46 refs, pp. xi+234, ISBN: 3-540-61695-0.

Publisher: Springer-Verlag, Berlin, Germany.

Author(s)

Kotz-D, Nieuwejaar-N.

Editor(s): Boszormenyi-L.





Author affiliation

Kotz, D., Nieuwejaar, N., Dept. of Comput. Sci., Dartmouth Coll., Hanover, NH, USA.

Abstract

As we gain experience with parallel **file** systems, it becomes increasingly clear that a **single** solution does not suit all applications. For example, it appears to be impossible to find a **single** appropriate **interface**, caching policy, **file** structure, or disk management strategy. Furthermore, the proliferation of **file** system **interfaces** and abstractions make applications difficult to port. We propose that the traditional functionality of parallel **file** systems be separated into two components: a fixed core that is standard on all platforms, encapsulating only primitive abstractions and **interfaces**, and a set of high level libraries to provide a variety of abstractions and application programmer **interfaces** (APIs). We present our current and next generation **file** systems as examples of this structure. Their features, such as a three dimensional **file** structure, strided read and write **interfaces**, and I/O node programs, are specifically designed with the flexibility and performance necessary to support a wide range of applications.

Descriptors

 APPLICATION-PROGRAM-INTERFACES;  FILE-ORGANISATION;  PARALLEL-PROGRAMMING;
 SOFTWARE-LIBRARIES.

Classification codes

C6150N Distributed-systems-software*;

C6120 File-organisation;

C6110P Parallel-programming;

C6150E General-utility-programs.

Keywords

parallel-file-systems-performance; **file-system-interfaces**; **fixed-core**; **primitive-abstractions**; **high-level-libraries**; **abstractions**; **application-programmer-interfaces**; **APIs**; **next-generation-file-systems**; **three-dimensional-file-structure**; **I/O-node-programs**.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper.

Publication year

1996.

Publication date

19960000.

Edition

1997001.

Copyright statement

Copyright 1997 IEE.

Full text available at



USPTO Full Text Retrieval Options

open url

☒ document 8 of 22 [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0005246521 20070101.

TitleFlexibility and performance of parallel **file** systems.**Source**

Operating Systems Review, {Oper-Syst-Rev-USA}, April 1996, vol. 30, no. 2, p. 63-73, 46 refs,
CODEN: OSRED8, ISSN: 0163-5980.
Publisher: ACM, USA.

Author(s)

Kotz-D, Nieuwejaar-N.

Author affiliation

Kotz, D., Nieuwejaar, N., Dept. of Comput. Sci., Dartmouth Coll., Hanover, NH, USA.

Abstract

Many scientific applications for high-performance multiprocessors have tremendous I/O requirements. As a result, the I/O System is often the limiting factor of application performance. Several new parallel **file** systems have been developed in recent years, each promising better performance for some class of parallel applications. As we gain experience with parallel computing, and parallel **file** systems in particular, it becomes increasingly clear that a **single** solution does not suit all applications. For example, it appears to be impossible to find a **single** appropriate **interface**, caching policy, **file** structure, or disk management strategy. Furthermore, the proliferation of **file**- system **interfaces** and abstractions make application portability a significant problem. The authors propose that the traditional functionality of parallel **file** systems be separated into two components: a fixed core that is standard on all platforms, encapsulating only primitive abstractions and **interfaces**, and a set of high-level libraries to provide a variety of abstractions and application-programmer **interfaces** (APIs). The authors think of this approach as the "RISC" of parallel **file-system** design. They present their current and next-generation **file** systems as examples of this structure. Their features, such as a three-dimensional **file** structure, strided read and write **interfaces**, and I/O-node programs, are specifically designed with the flexibility and performance necessary to support a wide range of applications.

Descriptors

APPLICATION-PROGRAM-INTERFACES; INPUT-OUTPUT-PROGRAMS; NETWORK-OPERATING-SYSTEMS; OPERATING-SYSTEM-KERNELS; OPERATING-SYSTEMS-COMPUTERS; SOFTWARE-PERFORMANCE-EVALUATION.

Classification codes

C61501 Operating-systems*;
C6150N Distributed-systems-software;
C6120 File-organisation.

Keywords

distributed-system-software; operating-system-kernel; **parallel-file**- system; software-performance; flexibility; high-performance-multiprocessor-system; I/O-requirement; input-output-system; fixed-core; high-level-library; **application-programmer-interface**; API; **parallel-file-system-design**; **three-dimensional-file-structure**; **strided-read-and-write-interface**.

Treatment codes

P Practical.

Language

English.

Publication typeJournal-paper.**Availability**

SICI: 0163-5980(199604)30:2L:63:FPPF; 1-Y.

Publication year

1996.

Publication date

19960400.

Edition

1996016.

Copyright statement

Copyright 1996 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options** [open url](#)☒ **document 9 of 22** [Order Document](#)**Inspec - 1898 to date (INZZ)****Accession number & update**

0005090704 20070101.

Title

Adding a data visualization tool to DEC FUSE.





Source

Digital Technical Journal, {Digit-Tech-J-USA}, 1995, vol. 7, no. 2, p. 20-33, 5 refs, CODEN: DTJOEL, ISSN: 0898-901X, USA.

Author(s)[Zaremba-D-A.](#)**Abstract**

Digital's data visualizer tool uses condensed **file views** to **display** thousands of lines of source code. These **displays** can include the output of many other tools. As part of the DEC FUSE programming environment, the tool helps software developers by providing capabilities for **displaying** large bodies of text with associated events or statistics. The data visualizer tool combines the results of other tools into a **single display**, keeps track of work items, and scales up to support large software projects.

Descriptors

 [DATA-VISUALISATION](#);  [GRAPHICAL-USER-INTERFACES](#);  [PROGRAMMING-ENVIRONMENTS](#);
 [SOFTWARE-TOOLS](#).

Classification codes

[C6180G Graphical-user-interfaces*](#);
[C6130B Graphics-techniques](#);
[C6115 Programming-support](#).

Keywords

data-visualization-tool; **condensed-file-views**; DEC-FUSE-programming-environment; software-developers; large-software-projects.

Treatment codes[P Practical](#).**Language**

English.

Publication type[Journal-paper](#).**Publication year**

1995.

Publication date

19950000.

Edition

1995042.

Copyright statement

Copyright 1995 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 10 of 22** [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0004957201 20070101.

Title

libscheme: Scheme as a C library.

Conference information

Proceedings of the USENIX Symposium on Very High Level Languages (VHLL), Sante Fe, NM, USA, 26-28 Oct. 1994.

Source

Proceedings of the USENIX Symposium on Very High Level Languages (VHLL), 1994, p. 7-19, 8 refs, pp. 226.

Publisher: USENIX Assoc, Berkeley, CA, USA.

Author(s)

[Benson-B-W--Jr.](#)

Abstract

Because of its small size and simplicity, Scheme is often seen as an ideal extension or scripting language. While there are many Scheme implementations available, their **interfaces** are often complex and can get in the way of using the implementation as part of a larger software product. The libscheme library makes the Scheme language available as a C library. Its **interface** is through a **single C header file** and it is easily extended with new primitive procedures, new primitive types, and new syntax. It is portable to any system that has an ANSI C compiler and to which Hans Boehm's popular conservative garbage collector (H. Boehm and M. Weiser, 1988) has been ported. It has been used to build a variety of special purpose data manipulation tools, and as an extension language for an Ethernet monitor.

Descriptors

☒ [C-LANGUAGE](#); ☒ [DATA-STRUCTURES](#); ☒ [HIGH-LEVEL-LANGUAGES](#); ☒ [SOFTWARE-LIBRARIES](#);
☒ [SOFTWARE-PORTABILITY](#).

Classification codes

[C6140D High-level-languages*](#);
[C6110B Software-engineering-techniques](#);
[C6120 File-organisation](#).

Keywords

libscheme; Scheme; C-library; scripting-language; extension-language; **single-C-header-file**; primitive-procedures; primitive-types; syntax; portable; ANSI-C-compiler; conservative-garbage-collector; special-purpose-data-manipulation-tools; Ethernet-monitor.

Treatment codes

[P Practical](#).

Language

English.

Publication type

[Conference-paper](#).

Publication year

1994.

Publication date

19940000.

Edition

1995020.

Copyright statement

Copyright 1995 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at [custom url](#) **USPTO Full Text Retrieval Options** [open url](#)

☒ **document 11 of 22** [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0004819366 20070101.

Title

Database management in C++.

Source

Dr. Dobb's Journal, {Dr-Dobb-s-J-USA}, Nov. 1994, vol. 19, no. 13, p. 36, 38, 40, 91-3, 0 refs, CODEN: DDJSDM, ISSN: 1044-789X, USA.








Author(s)

Sulger-A.

Abstract

I present a C++ class library which provides a **single interface** to multiple database **file** formats. In addition to freeing you from code duplication, this class structure allows your DBMS to support new data types such as those used with multimedia.

Descriptors

 ABSTRACT-DATA-TYPES;  C-LANGUAGE;  C-LISTINGS;  OBJECT-ORIENTED-DATABASES;  OBJECT-ORIENTED-LANGUAGES;  SUBROUTINES;  USER-INTERFACE-MANAGEMENT-SYSTEMS.

Classification codes

C6160J Object-oriented-databases*;
C6140D High-level-languages;
C6110J Object-oriented-programming;
C6120 File-organisation;
C6180 User-interfaces.

Keywords

C++-class-library; database-management; **interface**; multiple-database- **file-formats**; code-duplication; DBMS; data-types; multimedia.

Treatment codes

P Practical.

Language

English.

Publication type

Journal-paper.

Publication year

1994.

Publication date

19941100.

Edition

1994047.

Copyright statement

Copyright 1994 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 12 of 22** [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0004518476 20070101.

Title

Multiprocessor **file system interfaces**.

Conference information

Proceedings of the Second International Conference on Parallel and Distributed Information Systems (Cat. No.93TH0493-7), San Diego, CA, USA, 20-22 Jan. 1993.

Sponsor(s): IEEE; ACM.

Source

Proceedings of the Second International Conference on Parallel and Distributed Information Systems (Cat. No.93TH0493-7), 1993, p. 194-201, 31 refs, pp. xiv+272, ISBN: 0-8186-3330-1.

Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA.

Author(s)

Kotz-D.

Author affiliation

Kotz, D., Dept. of Math. & Comput. Sci., Dartmouth Coll., Hanover, NH, USA.

Abstract

The difficulties faced when using the conventional (UNIX-like) **interface** in parallel applications are described. Ways to extend the conventional **interface** to provide convenient access to the **file** for parallel programs, while retaining the traditional **interface** for programs that have no need to explicitly parallel **file** access, are described. The **interface** includes a **single** naming scheme, a multiopen operation, local and global **file** pointers, mapped **file** pointers, logical records, multifiles, and logical coercion for backward compatibility.

Descriptors

 FILE-ORGANISATION;  MULTIPROCESSING-PROGRAMS;  PARALLEL-PROGRAMMING.

Classification codes

C6150N Distributed-systems-software*;

C6110P Parallel-programming;

C6120 File-organisation.

Keywords

Unix; parallel-applications; parallel-programs; **parallel-file-access**; naming-scheme; multiopen-operation; **global-file-pointers**; **mapped-file**-pointers; logical-records; multifiles; logical-coercion; backward-compatibility.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper.

Availability

CCCC: 0 8186 3330 1/93/\$03.00.

Digital object identifier

10.1109/PDIS.1993.253093.

Publication year

1993.

Publication date

19930000.

Edition

1993043.

Copyright statement

Copyright 1993.IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at **USPTO Full Text Retrieval Options** [open url](#)

☒ **document 13 of 22** [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0004185445 20070101.

TitleA **display** processor.**Source**

Optoelectronics Instrumentation and Data Processing, {Optoelectron-Instrum-Data-Process-USA}, 1991, no. 2, p. 36-41, CODEN: OIDPE4, ISSN: 8756-6990, USA.

Translation from: Avtometriya, {Avtometriya-Russia}, 1991, no. 2, p. 32-7, CODEN: AVMEBI, ISSN: 0320-7102.

Country of publication: Russia.

Author(s)Kablukov-V-I, Kozlachkov-V-A, Korshever-I-I, Pavlov-S-A, Teslenko-K-V, Shadrin-M-Yu.**Abstract**

A **single-board display** processor (DP) of intermediate capacity is described, oriented toward incorporation in a computing system based on a Q-bus. The main characteristic features of the processor are: vector representation of input data, a wide color gamut, the possibility of performing various operations on input images, a large virtual **display** field, high rate of block transfers, and adaptable interaction with a control computer. The DP organization and operation, the principle of its interaction with the control computer, and its **display file** structure are described.

DescriptorsCOMPUTER-GRAPHIC-EQUIPMENT; COMPUTER-INTERFACES; COMPUTERISED-PICTURE-PROCESSING; SIGNAL-PROCESSING-EQUIPMENT.**Classification codes**C5540 Terminals-and-graphic-displays*;C5610S System-buses;C5260 Digital-signal-processing.**Keywords**

specialised-peripheral-processor; microprogram-control; interactive-computer-graphics; **single-board-display-processor**; intermediate-capacity; Q-bus; vector-representation; input-data; wide-color-gamut; **large-virtual-display-field**; high-rate-of-block-transfers; adaptable- interaction; control-computer; **display-file-structure.**

Treatment codesP Practical.**Language**

English.

Publication typeJournal-paper.**Availability**

CCCC: 8756-6990/91/\$20.00.

Publication year

1991.

Publication date

19910000.

Edition

1992031.

Copyright statement

Copyright 1992 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at [custom url](#) [USPTO Full Text Retrieval Options](#) [open url](#)☒ **document 14 of 22** [Order Document](#)**Inspec - 1898 to date (INZZ)****Accession number & update**

0003918302 20070101.

Title

A look at remote procedure calls.

Source

BYTE, {BYTE-USA}, May 1991, vol. 16, no. 5, p. 309-12, 314, 316, 317, 384-5, 0 refs, CODEN: BYTEDJ, ISSN: 0360-5280, USA.

Author(s)Mallett-M.**Abstract**

The method of remote procedure calls does for procedure-level programming what the Network **File** System does for data sharing: it attempts to hide the **view** of the total system as discrete components. Where NFS abstracts the **view** of data through a **single file-system** name space, RPC abstracts the **view** of client/server requests in the programming domain: the procedure name space. The author describes the technique with examples in C.

DescriptorsC-LISTINGS; PROGRAMMING.**Classification codes**C6110 Systems-analysis-and-programming*.**Keywords**

remote-procedure-calls; procedure-level-programming; RPC; procedure-name-space.

Treatment codesP Practical.**Language**

English.

Publication typeJournal-paper.**Publication year**

1991.

Publication date

19910500.

Edition

1991015.

Copyright statement

Copyright 1991 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 15 of 22** Order Document**Inspec - 1898 to date (INZZ)****Accession number & update**

0003799391 20070101.

TitleA concurrent **file** system for a highly parallel mass storage subsystem.**Conference information**

Proceedings of the Fourth Conference on Hypercubes, Concurrent Computers and Applications, Monterey, CA, USA, 6-8 March 1989.

Sponsor(s): D.O.E; US Air Force; NASA.

Source

Proceedings of the Fourth Conference on Hypercubes, Concurrent Computers and Applications, 1989, p. 155-60 vol.1, 4 refs, pp. 2 vol. xiv+1362.

Publisher: Golden Gate Enterprises, Los Altos, CA, USA.




Author(s)Pierce-P.**Author affiliation**

Pierce, P., Intel Sci. Comput., Beaverton, OR, USA.

Abstract

The iPSC/2 I/O subsystem consists of many small disks served by many parallel I/O channels. This is achieved by connecting groups of disks to I/O nodes, each of which has full access to the hypercube interconnect of the computational nodes. All nodes in the system run the NX/2 node executive, which provides message passing and process management services. The Concurrent **File System** (CFS) software consists of specialized processes which run on the I/O nodes, and **file** management code which runs in conjunction with the application on compute nodes. The familiar Unix programming **interface** provides a simple **single file system view**, underneath which the CFS transparently manages parallelism in distribution and concurrent access to **files**. Multiple disk block caches and read-ahead schemes provide high performance.

Descriptors

 HYPERCUBE-NETWORKS;  PARALLEL-PROCESSING;  STORAGE-MANAGEMENT.

Classification codes

C6150J Operating-systems*;
C6120 File-organisation;
C5440 Multiprocessing-systems.

Keywords

parallel-distributed-memory-architectures; **concurrent-file-system**; highly-parallel-mass-storage-subsystem; iPSC/2-I/O-subsystem; hypercube-interconnect; NX/2-node-executive; **file-management-code**; **Unix-programming-interface**; **single-file-system-view**; CFS- transparently; concurrent-access; read-ahead-schemes.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper.

Publication year

1989.

Publication date

19890000.

Edition

1991003.

Copyright statement

Copyright 1991 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 16 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0003768130 20070101.

Title

Tools for remote computing in accelerator control.

Conference information

International Conference on Accelerator and Large Experimental Physics Control Systems, Vancouver, BC, Canada, 30 Oct.-3 Nov. 1989.

Sponsor(s): TRIUMF; Eur. Phys. Soc; IEEE.

Source

Nuclear Instruments & Methods in Physics Research Section A (Accelerators Spectrometers Detectors and Associated Equipment), {Nucl-Instrum-Methods-Phys-Res-A-Accel-Spectrom-Detect-Assoc-Equip-Netherlands}, 1 Aug. 1990, vol. A293, no. 1-2, p. 225-30, 14 refs, CODEN: NIMAER, ISSN: 0168-9002, Netherlands.

Author(s)

Anderssen-P-S, Frammery-V, Wilcke-R.

Author affiliation

Anderssen, P.S., Frammery, V., CERN, Geneva, Switzerland.

Abstract

In modern accelerator control systems, the intelligence of the equipment is distributed in the geographical and the logical sense. Control processes for a large variety of tasks reside in both the equipment and the control computers. Hence successful operation hinges on the availability and reliability of the communication infrastructure. The computers are interconnected by a communication system and use remote procedure calls and message passing for information exchange. These communication mechanisms need a well-defined convention, i.e. a protocol. They also require flexibility in both the setup and changes to the protocol specification. The Network Compiler is a tool which provides the programmer with a means of establishing such a protocol for his application. Input to the Network Compiler is a **single Interface Description File** provided by the programmer. This **file** is written according to a grammar, and completely specifies the interprocess communication **interfaces**. Passed through the Network Compiler, the **Interface Description File** automatically produces the additional source code needed for the protocol. Hence the programmer does not have to be concerned about the details of the communication calls. Any further additions and modifications are made easy, because all the information about the **interface** is kept in a **single file**.

Descriptors

 CONTROL-SYSTEMS;  PARTICLE-ACCELERATOR-ACCESSORIES;  PHYSICS-COMPUTING.

Classification codes

A2915 Electrostatic-and-linear-particle-accelerators*;
A2920 Cyclic-accelerators-and-storage-facilities;
C7320 Physics-and-chemistry-computing*;
C3210P Control-systems.

Keywords

remote-computing; accelerator-control; control-systems; communication-infrastructure; remote-procedure-calls; message-passing; information-exchange; Network-Compiler; **Interface-Description-File**; interprocess- **communication-interfaces**; additional-source-code.

Treatment codes

A Application;
P Practical.

Language

English.

Publication type

Conference-paper; Journal-paper.

Availability

CCCC: 0168-9002/90/\$03.50.

Publication year

1990.

Publication date

19900801.

Edition

1991001.

Copyright statement

Copyright 1991 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 17 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0003351440 20070101.

Title

Electronic charts-at the heart of the 21st century navigation.

Conference information

IEEE PLANS '88 Position Location and Navigation Symposium Record - 'Navigation into the 21st Century' (IEEE Cat. No.88CH2675-7), Orlando, FL, USA, 29 Nov.-2 Dec. 1988.
Sponsor(s): IEEE.

Source

IEEE PLANS '88 Position Location and Navigation Symposium Record - 'Navigation into the 21st Century' (IEEE Cat. No.88CH2675-7), 1988, p. 88-94, 0 refs, pp. 559.
Publisher: IEEE, New York, NY, USA.

Author(s)

Rogoff-M.

Abstract

The electronic chart is a facsimile or replica of a nautical chart as **displayed** on an electronic screen. In most instances the **display** is created from a digital **file** that is produced by manually digitizing the curves and contours of a paper chart. It is suggested that, in maritime applications, the chart's presence will enable the integration of radar, positioning, and communications equipment into a **single display**; this will enhance the value of these systems, and will have a strong influence on the structure of the marine electronics industry. One of the problems of acceptance of electronic charts is not technical, but is legal and regulatory. It is concluded that international and national standards and rules will have to be adopted before there is widespread purchase and use of these systems.

Descriptors

 CARTOGRAPHY;  COMPUTERISED-NAVIGATION;  MARINE-SYSTEMS;  STANDARDS.

Classification codes

C3360J Marine-system-control*;
C7490 Computing-in-other-engineering-fields;
C7420 Control-engineering-computing.

Keywords

international-standards; electronic-chart; facsimile; nautical-chart; **digital-file**; curves; contours; maritime-applications; radar; positioning; communications; marine-electronics-industry; national-standards.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper.

Availability

CCCC: CH2675-7/88/0000-0088\$01.00.

Digital object identifier

10.1109/PLANS.1988.195470.

Publication year

1988.

Publication date

19880000.

Edition

1989009.

Copyright statement

Copyright 1989 IEE.

(c) 2007 The Institution of Engineering and Technology

☒ **document 18 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0002967069 20070101.

Title

An empirical study of software **interface** faults-an update.

Conference information

Proceedings of the Twentieth Hawaii International Conference on System Sciences 1987, Kailua-Kona, HI, USA, 6-9 Jan. 1987.

Sponsor(s): Univ. Hawaii; Univ. SW Louisiana; ACM; IEEE.

Source

Proceedings of the Twentieth Hawaii International Conference on System Sciences 1987, 1987, p. 113-26 vol.2, 10 refs, pp. 3 vol. (xviii +781+xxxvi+555+xvii+510).

Publisher: Hawaii Int. Conference Syst. Sci, Honolulu, HI, USA.

Author(s)

Perry-D-E.

Editor(s): Stohr-E-A, Hoevel-L, Haynes-L, Chu-Y, Speckhard-A, Shriver-B-D, Grams-R-R, Sprague-R-H-Jr.

Author affiliation

Perry, D.E., AT&T Bell Labs., Murray Hill, NJ, USA.

Abstract

For original article see Proc. of the new directions in Computing conf., Trondheim, Norway p.32-7 (Aug 1985). The authors extend the data analysis to include faults that affected only one code **file**. They compare the differing characterizations of **interface** faults provided by the two different data sets and then combine the data to get an overall characterization. They find that many of the problems can be attributed to methodology and that implementation errors far outweigh design errors in this data.

Descriptors

 SOFTWARE-RELIABILITY.

Classification codes

C6110 Systems-analysis-and-programming*.

Keywords

single-file-interface-faults; software-interface-faults; data- analysis; implementation-errors; design-errors.

Treatment codes

X Experimental.

Language

English.

Publication type

Conference-paper.

Availability

Available from: Western Periodicals, North Hollywood, CA, USA.

Publication year

1987.

Publication date

19870000.

Edition

1987019.

Copyright statement

Copyright 1987 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at



USPTO Full Text Retrieval Options

open url

☒ **document 19 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0001880665 20070101.

Title

A distributed UNIX system based on a virtual circuit switch.

Conference information

Proceedings of the Eighth Symposium on Operating Systems Principles, Pacific Grove, CA, USA, 14-16 Dec. 1981.

Source

Operating Systems Review, {Oper-Syst-Rev-USA}, Dec. 1981, vol. 15, no. 5, p. 160-8, 27 refs, CODEN: OSRED8, ISSN: 0163-5980, USA.

Author(s)

Luderer-G-W-R, Che-H, Haggerty-J-P, Kirslis-P-A, Marshall-W-T.




Author affiliation

Luderer, G.W.R., Che, H., Haggerty, J.P., Bell Labs., Murray Hill, NJ, USA.

Abstract

The popular UNIX operating system provides time-sharing service on a **single** computer. This paper reports on the design and implementation of a distributed UNIX system. The new operating system consists of two components: the S-UNIX subsystem provides a complete UNIX process environment enhanced by access to remote **files**; the F-UNIX subsystem is specialized to offer remote **file** service. A system can be configured out of many computers which operate either under the S-UNIX or the F-UNIX operating subsystem. The **file** servers together present the **view** of a **single** global **file** system. A **single-service view** is presented to any user terminal connected to one of the S-UNIX subsystems. Computers communicate with each other through a high-bandwidth virtual circuit switch. Small front-end processors handle the data and control protocol for error and flow-controlled virtual circuits. Terminals may be connected directly to the computers or through the switch. Operational since early 1980, the system has served as a vehicle to explore virtual circuit switching as the basis for distributed system design. The performance of the communication software has been a focus of the work. Performance measurement results are presented for user process level and operating system driver level data transfer rates, message exchange times, and system capacity benchmarks. The architecture offers reliability and modularly growable configurations. The communication service offered can serve as a foundation for different distributed architectures.

Descriptors

 COMPUTER-NETWORKS;  OPERATING-SYSTEMS-COMPUTERS;  VIRTUAL-MACHINES.

Classification codes

B6210L Computer-communications*;
C5620 Computer-networks-and-techniques*;
C6150J Operating-systems.

Keywords

computer-networks; virtual-machines; distributed-UNIX-system; virtual-circuit-switch; operating-system; time-sharing-service; S-UNIX; **remote-files**; F-UNIX; **remote-file-service**; front-end-processors; control-protocol; flow-controlled-virtual-circuits; distributed-system-design; communication-software.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper; Journal-paper.

Publication year

1981.

Publication date

19811200.

Edition

1982007.

Copyright statement

Copyright 1982 IEE.

(c) 2007 The Institution of Engineering and Technology

Full text available at



IEEE

USPTO Full Text Retrieval Options

[open url](#)

☒ **document 20 of 22** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0001815368 20070101.

Title

The effect of airborne contamination and environmental conditions on the head/disk **interface** in a **single flexible disk file**.

Conference information

1981 International Magnetism Conference, Grenoble, France, 12-15 May 1981.

Sponsor(s): IEEE.

Source

IEEE Transactions on Magnetism, {IEEE-Trans-Magn-USA}, Nov. 1981, vol. MAG-17, no. 6, p. 2751-3, 1 refs, CODEN: IEMGAQ, ISSN: 0018-9464, USA.

Author(s)

Feliss-N-A, Talke-F-E.



Author affiliation

Feliss, N.A., Talke, F.E., IBM Corp., San Jose, CA, USA.

Abstract

The authors investigated the effect of airborne contamination on the head/flexible disk **interface** by injecting particles of known size distribution into the neighborhood of the rotating disk. Changes in the head/disk fringe pattern are monitored as a function of particle size using a high resolution video camera coupled to an on-line automatic data acquisition system. In addition, the effect of severe environmental conditions on the head/disk **interface** is studied for selected temperatures and humidities. All measurements of the head /disk spacing were performed using clear mylar disks. Similar trends of the results are observed for oxide coated disks.

Descriptors

 MAGNETIC-DISC-AND-DRUM-STORAGE;  MAGNETIC-HEADS.

Classification codes

C5320C Storage-on-moving-magnetic-media*;

C5320 Digital-storage.

Keywords

environmental-conditions; **single-flexible-disk-file**; airborne-contamination; **head/flexible-disk-interface**; head/disk-fringe-pattern; particle-size; temperatures; humidities; head/disk-spacing; oxide-coated-disks.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-paper; Journal-paper.

Publication year

1981.

Publication date

19811100.

Edition

1982003.

Copyright statement

Copyright 1982.IEE.

(c) 2007 The Institution of Engineering and Technology

next
documents

save

locally as: PDF document



search strategy: do not include the search strategy



order

copy to
Clipboard

Top - News & FAQs - Dialog

© 2007 Dialog